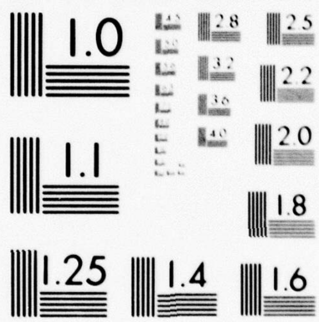
AD-A060 821 STANFORD UNIV CALIF DEPT OF OPERATIONS RESEARCH F/G 12/1
LEAST-INDEX RESOLUTION OF DEGENERACY IN QUADRATIC PROGRAMMING.(U)
FEB 78 Y CHANG, R W COTTLE N00014-75-C-0267
UNCLASSIFIED TR-78-3 NL

1 OF 1
AD
A060 821

END
DATE
FILMED
1-79
DDC




MICROCOPY RESOLUTION TEST CHART

LEVEL II

(12)

DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305



(6)

# LEAST-INDEX RESOLUTION OF DEGENERACY IN QUADRATIC PROGRAMMING

by

(10) Yow-Yieh Chang ~~and~~ Richard W. Cottle

TECHNICAL REPORT 78-3
(11) ~~February~~ 1978

(12) 22 p.



(9) Technical rept.,

(14) TR-78-3

(15) N00014-75-C-0267, NSF-MCS76-81259

Research and reproduction of this report were partially supported by National Science Foundation Grant MCS76-81259; and the Office of Naval Research Contract N00014-75-C-0267.



402 766



JOB

LEAST-INDEX RESOLUTION OF DEGENERACY IN QUADRATIC PROGRAMMING

by

Yow-Yieh CHANG and Richard W. COTTLE
Department of Operations Research
Stanford University

ABSTRACT

In this study, we combine least-index pivot selection rules with Keller's algorithms for quadratic programming to obtain a finite method for processing degenerate problems.

## 1. Introduction.

Degeneracy is a theoretically troublesome phenomenon in the analysis of simplicial methods for linear programming although, according to folklore, it is not a serious impediment to practical computation [7, p. 231]. For the record, a basic solution to a linear programming problem is said [7, p. 81] to be degenerate if at least one of the basic (i.e. dependent) variables equals 0. Even if the initial basic solution is non-degenerate, an adjacent extreme point algorithm such as Dantzig's simplex method may select degenerate basic solutions as a consequence of ties in the minimum ratio test used to determine the exiting basic variable. This in turn can lead to the phenomenon known as circling (alias cyling): a sequence of bases which (after finitely many steps) repeats itself. Degeneracy per se is not the problem; but when it is present, circling is a possibility and must be avoided if the simplex method and procedures like it are to work and be finite. The "degeneracy problem" refers to the difficulties associated with circling.

Until quite recently, the theoretical procedures available for handling the degeneracy problem were Charnes' perturbation technique [ 3 ], the lexicographic method of Dantzig, Orden and Wolfe [ 8 ], the ad hoc method of Wolfe [15]. We shall not attempt to review these technical procedures here, but simply remark that their actual implementations are commonly believed to entail storage requirements and computational effort out of all proportion to the need for them. In 1976, Bland [ 2 ] announced a finite version of the simplex method based upon a double least-index pivot selection rule. In Bland's method, the entering basic variable is chosen as the candidate with the smallest index. The variable it replaces (if any) is determined by the usual minimum ratio test with ties broken according to the least-index rule. The method seems so natural, it is difficult to understand why it was not published earlier.

At this point, Bland's contribution is of greater importance to linear programming theory than practice. A study by Avis and Chvátal [ 1 ] shows that the method (in the form sketched above) is less efficient than those which choose the entering variable by paying more attention to decreasing the objective function. Avis and Chvátal found, however, that prior arrangement of the variables may help to bring about <u>greater</u> efficiency as compared with the other methods. Another possible shortcoming of Bland's method may lie in its disregard of the magnitude of the pivot entry. Recognizing these objections, we regard Bland's least-index rule as a significant contribution to LP theory and hope that it will develop into a viable practical approach as well.

In this paper we demonstrate that Bland's least-index pivot selection rules can be applied to handle the degeneracy problem in the Dantzig/van de Panne-Whinston/Keller algorithm for quadratic programming. (See [6], [7], [13], [9].) In a nutshell, what is shown is that by using Bland's least-index rules in Keller's algorithm, one can prevent circling and reach one of the terminal forms after finitely many pivot steps. From this one either has knowledge that the objective function is unbounded below or else a solution of the Kuhn-Tucker conditions.

## 2. Quadratic programming

In this section we consider the application of Bland's least index rule to the symmetric Dantzig/van de Panne-Whinston/Keller quadratic programming algorithm. This procedure in its most general form (given by Keller) aims to find a local minimum in a nonconvex quadratic programming problem.

### 2.1 Statement of the problem.

We express the quadratic programming problem in the form

$$\text{(1)}\qquad \begin{aligned} \text{minimize}\quad & Q(x) = c^T x + \tfrac{1}{2} x^T D x \\ \text{subject to}\quad & Ax \le b \\ & x \ge 0 \end{aligned}$$

The matrix D is always (without loss of generality) assumed symmetric. When we speak of convex quadratic programming, we refer to the case where D is positive semi-definite; but for the moment this is not assumed.

The approach taken in the Dantzig/van de Panne-Whinston/Keller algorithms for solving (1) is to find a solution of its Kuhn-Tucker conditions, i.e. its necessary conditions of optimality, namely

$$\text{(2)}\qquad \begin{aligned} & u = c + Dx + A^T y \\ & v = b - Ax \\ & x \ge 0\ ,\ y \ge 0\ ,\ u \ge 0\ ,\ v \ge 0 \\ & x^T u = 0\ ,\ \ y^T v = 0 \end{aligned}$$

In the convex quadratic programming case these conditions are also sufficient for the optimality of x. In the nonconvex case, they are not in general sufficient, even for a local minimum.

The method of Keller (which extends those of Dantzig and van de Panne-Whinston) is capable of finding a local minimum of a nondegenerate nonconvex QP in finitely many steps. This is done by obtaining a suitable solution of (2). We show here that by using a natural adaptation of Bland's least-index rule we can dispense with the nondegeneracy assumption and find a solution of (2), again in finitely many steps.

## 2.2 Bland's least-index rule for LP.

In 1976, R.G. Bland [ 2 ] showed that a certain simple and natural pivot selection rule never leads to circling. His pivoting rule is a refinement of the simplex rule obtained by imposing the following restrictions:

(a) among all the candidates to enter the basic set of variables, select the nonbasic variable having the lowest index.

(b) among all candidates to leave the basic set, select the basic variable having the lowest index.

It is well to understand what is meant by these statements. We imagine the linear program to be expressed in the canonical simplex tableau consisting of $m + 1$ linearly independant rows and $n + 2$ columns;

| Basic variables | $x_0$ | $x_1$ | $\cdots$ | $x_n$ | $1$ |
|---|---|---|---|---|---|
| $x_{B_0}$ | $1$ | $a_{01}$ | $\cdots$ | $a_{0n}$ | $b_0$ |
| $x_{B_1}$ | $0$ | $a_{11}$ | $\cdots$ | $a_{1n}$ | $b_1$ |
| $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ | $\vdots$ |
| $x_{B_m}$ | $0$ | $a_{m1}$ | $\cdots$ | $a_{mn}$ | $b_m$ |

It is assumed that $B_0 = 0$ , $1 \leq B_i \leq n$ for $i = 1, \ldots, m$ , and the matrix formed from the columns headed by $x_{B_0}$, $x_{B_1}$ , ..., $x_{B_m}$ constitutes an identity matrix of order $(m + 1)$. In particular, the columns associated with $x_{B_i}$ has a 1 in row i and 0 elsewhere.

The least-index selection rule for choosing the incoming variable is clear although it differs from the customary one. We want to stress that in choosing the exiting basic variable one uses the usual candidacy rule and breaks ties according to the least index of the affected basic variables rather than by the least row number of such variables. To be precise, suppose $x_s$ is the incoming variable. Then the exiting basic variable (if any) is the one whose index (subscript) satisfies

$$B_r = \min\left\{B_\ell : \ a_{\ell s} > 0 \text{ and } \frac{b_\ell}{a_{\ell s}} = \min\left\{\frac{b_i}{a_{is}} : \ a_{is} > 0\right\}\right\} \tag{3}$$

The pivot element is then $a_{rs}$. The important point is that the minimization in (3) is over $B_\ell$ rather than $\ell$.

### 2.3 Keller's algorithm.

In describing Keller's algorithm, it is convenient to introduce a change of notation in (2). Suppose $A \in R^{m\times n}$. Initially, we define $I = \{1, \ldots, n\}$ , $J = \{n + 1, \ldots, n + m\}$ , and then put

$$x_I = x\ , \quad x_J = v\ , \quad y_I = u\ , \quad y_J = y \tag{4}$$

Accordingly, (2) can be replaced by the bisymmetric schema:

(5)

$$
\begin{array}{r|c|cc|}
 & 1 & x_I & y_J \\
\hline
\theta = & 0 & c^T & -b^T \\
\hline
y_I = & c & D & A^T \\
x_J = & b & -A & 0 \\
\hline
\end{array}
$$

The variable $\theta$ is related to the objective function through the equation

(6) $$\theta = 2Q - x_I^T y_I - x_J^T y_J$$

This equation is a consequence of two other equations represented in (5).

Actually, Keller's method works with the slightly more general schema

(7)

$$
\begin{array}{r|c|cc|}
 & 1 & x_I & y_J \\
\hline
\theta = & 2\kappa & c^T & -b^T \\
\hline
y_I = & c & D & A^T \\
x_J = & b & -A & E \\
\hline
\end{array}
$$

in which E is <u>symmetric</u> and positive <u>semi-definite</u> and $\kappa$ is a constant. (In our original schema (5) we have $E = 0$ and $\kappa = 0$.) The most important reason for working with such a schema is that one encounters this type of format in subsequent principal pivotal transforms of (5), so one might as well regard it as given from the start. A second rationale for (7) is that it corresponds to the one used for symmetric dual quadratic programs [ 4 ]. In such a schema, we regard the variables $y_i$ $(i \in I)$ and $x_j$ $(j \in J)$ as basic. The variables $x_i$ $(i \in I)$ and $y_j$ $(j \in J)$ are then non-basic. It is helpful to notice that the entries in the schema can be considered as partial derivatives. For example

$$\partial y_i / \partial x_j = d_{ij}$$

It is assumed that $b \geqq 0$. This amounts to a (primal) feasibility condition. In a given instance, if $b$ is not nonnegative one can as an initialization step execute the Phase I procedure of LP to make it so or else determine the infeasibility of the constraints. The remaining steps of the original method for the nondegenerate case run as follows:

Step 1. If all basic y-variables are nonnegative, stop. A local minimum has has been found. Otherwise choose a negative y-component, say $y_s$, as the "distinguished variable".

Step 2. If $\partial y_s / \partial x_s \leqq 0$ go to Step 3. Otherwise, determine the "blocking variable", i.e. the basic variable (either $y_s$ or $x_j$, $j \in J$) which reaches the value 0 first as $x_s$ is increased from 0.

a. If $y_s$ is the blocking variable, perform the in-pivot $\langle y_s, x_s \rangle$ by which $y_s$ and $x_s$ are exchanged. Replace $J$ by $J \cup \{s\}$ and $I$ by $I - \{s\}$. Return to Step 1.

b. If $x_t$ is the blocking variable and $\partial x_t / \partial y_t > 0$ then perform the out-pivot $\langle x_t, y_t \rangle$ by which $x_t$ and $y_t$ are exchanged. Replace $J$ by $J - \{t\}$ and $I$ by $I \cup \{t\}$. Repeat Step 2 (with $x_s$ still the driving variable). If $\partial x_t / \partial y_t = 0$, then perform the exchange pivot $\langle x_t, x_s \rangle$, $\langle y_s, y_t \rangle$ through which $x_t$ and $x_s$ are exchanged and $y_s$ and $y_t$ are exchanged. Replace $J$ by $(J - \{t\}) \cup \{s\}$ and $I$ by $(I - \{s\}) \cup \{t\}$. Return to Step 1.

Step 3. Determine whether $x_s$ is blocked. If $\partial x_j / \partial x_s \geq 0$ for all $j \in J$ then $x_s$ is unblocked. Stop, since $Q \to -\infty$. Otherwise $x_s$ is blocked. If $\partial x_t / \partial y_t > 0$, perform the out-pivot $\langle x_t, y_t \rangle$ ; replace $J$ by $J - \{t\}$ and $I$ by $I \cup \{t\}$. Return to Step 2. If $\partial x_t / \partial y_t = 0$, perform the exchange pivot $\langle x_t, x_s \rangle$ , $\langle y_s, y_t \rangle$ ; replace $J$ by $(J - \{t\}) \cup \{s\}$ and $I$ by $(I - \{s\}) \cup \{t\}$. Return to Step 1.

We do not intend to justify the method or even discuss it at length. For this one should see [ 9]. However, we do wish to draw the reader's attention to some of its salient features.

1. The nondegeneracy assumption implies that there is always at most one blocking variable.
2. The method uses only principal pivots of order 1 (in-pivots and out-pivots) or order 2 (exchange pivots).
3. The property of bisymmetry is preserved by principal pivoting (regardless of the order). This is proved in [14].
4. The nonnegativity of the x-variables is preserved throughout the procedure.
5. Given the current index set $J$ the principal submatrix $\bar{E}$ corresponding to the rows and columns indexed by $J$ is always positive semi-definite and its nullity equals that of the original, E.
6. According to quadratic programming theory, a Kuhn-Tucker point is a local

minimum in the nondegenerate case provided $\bar{E}$ is positive semi-definite. However, in the degenerate case, more than just the positive semi-definiteness of $\bar{E}$ is needed. In effect, a type of copositivity condition is required. (See [10] and [11]). Unfortunately Keller's method makes no provision for this; hence we make no claim that the modified Keller method described below actually yields a local minimum in the degenerate case. Nevertheless, the method leads in finitely many steps to an indication of unboundedness or else to a solution of the Kuhn-Tucker conditions, and in the case of convex quadratic programming, this is enough for global optimality.

## 2.4 Finiteness of Keller's method with the least index rule.

We now give a modification of Keller's algorithm and prove its finiteness without recourse to a nondegeneracy assumption. To accomplish this, we introduce a refinement of Keller's algorithm which imposes the following double least-index rule:

(i) In Step 1, choose the distinguished variable $y_s$ so that

$$s = \min \{i \in I: \; y_i < 0\}$$

(ii) In Steps 2(b) and 3, choose the blocking variable $x_t$ so that

$$t = \min \left\{ j \in J: -a_{js} < 0 \text{ and } \frac{b_j}{a_{js}} = \min \left\{ \frac{b_k}{a_{ks}} : k \in J, \; -a_{ks} < 0 \right\} \right\}$$

In Step 2, there could be a "tie" between $y_s$ and $x_t$ for blocking. The statement of Step 2 is intended to mean that under these circumstances, one should choose $y_s$ as the blocking variable. This would decrease the objective function value.

In Keller's method, each return to Step 1 completes a major cycle.

Lemma 1. Each major cycle of Keller's algorithm (with or without the least index rule) is finite.

Proof. The pivots in Keller's method are of three types: in-pivots, out-pivots, and exchange-pivots. Both in-pivots and exchange-pivots lead back to Step 1, hence each completes a major cycle. Each out-pivot reduces the cardinality of $J$ by 1, so there can be only finitely many out-pivots within a major cycle. □

Lemma 2. If circling occurs in Keller's algorithm, then during circling, only exchange-pivots are used.

Proof. By Lemma 1, if circling occurs, there must be infinitely many returns to the same complementary basis in terms of which the schema is uniquely determined. Keller shows that regardless of whether degeneracy is present, the value of $\theta$ is nonincreasing under all steps of the algorithm. Hence during circling, $\theta$ must be fixed. This implies there can be no in-pivots during circling, for these always decrease the value of $\theta$. Now it remains to show that during circling, there are no out-pivots. To see this we note that each out-pivot increases the cardinality of the index set $I$ (corresponding to the basic y-variables).

Since exchange-pivots do not affect the cardinality of $I$ and in-pivots cannot occur by our previous argument, there can be no out-pivots during circling. □

Suppose circling occurs during the execution of Keller's algorithm. We know that each major cycle must be finite, so the only remaining possibility is that there are infinitely many major cycles (returns to Step 1). A circle begins at a complementary schema, a principal pivotal transform of the original schema (7). Since non-terminal principal transforms of (7) share its properties (bisysmmetry, positive semi-definiteness of E, $b \geq 0$, $c \ngtr 0$), we may assume for simplicity that the circling starts at (7). The circling consists of a sequence of (at least two) exchange pivots returning to (7). Assuming the circling starts at (7) we focus attention on the variables actually exchanged.

<u>Definition</u>. Let

$$K = \{j \in J: \ x_j \ \text{ becomes nonbasic during circling}\}$$

Accordingly, by deleting from (7) those rows and columns j such that $j \notin K$, we obtain a <u>subschema</u> with much the same character, namely

(8)

| | 1 | $x_I$ | $y_K$ |
|---|---|---|---|
| $\theta =$ | $2\kappa$ | $c^T$ | $-b_K^T$ |
| $y_I =$ | $c$ | $D$ | $A_{K\cdot}^T$ |
| $x_K =$ | $b_K$ | $-A_{K\cdot}$ | $E_{KK}$ |

<u>Caution</u>: In (8) and the lemma below, we indulge in an abuse of notation. The symbols $b_K$ , $A_{K.}$ , and $E_{KK}$ refer to those parts of b, A, and E <u>corresponding</u> to the basic variables $x_K$.

<u>Lemma 3</u>. Under the assumptions made above,

$$b_K = 0 \quad \text{and} \quad E_{K.} = 0 \ , \ E_{.K} = 0 \tag{9}$$

Proof. Let $x_s$ be the first driving variable. Since this must lead to an exchange-pivot, let $x_t$ be the blocking variable. Since we must perform an exchange-pivot (rather than an out-pivot) we must have $e_{tt} = 0$. This implies

$$E_{t.} = 0 \ , \ E_{.t} = 0$$

because E is symmetric and positive semi-definite. As a result of an exchange-pivot, $2\kappa$ becomes

$$2\kappa + b_t \left[ \frac{c_s}{a_{ts}} + \frac{1}{a_{ts}} \left( c_s + \frac{b_t d_{ss}}{a_{ts}} \right) \right]$$

Each of the summands within the square brackets is negative. Since $b_t$ is nonnegative and $\theta$ does not change during circling, we must have $b_t = 0$. Under these circumstances, the entries of $b_K$ and $E_{KK}$ are not affected by the exchange-pivot. Since each row indexed by $k \in K$ is involved in an exchange-pivot, the argument just given applies and the proof is complete. □

Now we come to our result on quadratic programming.

Theorem 1. With the double least-index rule, Keller's algorithm is finite.

Proof. If the method is not finite, there is a subschema of the form

(10)
$$\begin{array}{c|c|cc|} & 1 & x_I & y_K \\ \hline \theta = & 2\kappa & c^T & 0 \\ \hline y_I = & c & D & A_{K\cdot}^T \\ x_K = & 0 & -A_{K\cdot} & 0 \\ \hline \end{array}$$

in which circling occurs. All of the transformations of this subschema arise as exchange-pivots. The matrix $D$ has no affect on the transforms of $c$ and $A_{K\cdot}$. Indeed $c$ and $A_{K\cdot}$ would be transformed the same way if $D$ were the zero matrix. But when $D = 0$, we are just doing Bland's refinement of the simplex method which is finite. □

Corollary. If circling occurs when Keller's method (without degeneracy precautions) is applied to the quadratic program ( 1 ), then $m \geq 2$ and $m + n \geq 6$. The bounds are sharp.

Proof. The arguments given above imply the existence of a linear program based on a subset of the variables in which circling occurs. But Marshall and Suurballe [12] have shown that such an LP must have at least two equations and at least six variables. Hence, $m \geq 2$ and $m + n \geq 6$.

The bounds given by Marshall and Suurballe are sharp, so they must be sharp here too. □

<u>Example</u>. The following minimal example is a modification of one given in [12].

| | 1 | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y_5$ | $y_6$ |
|---|---|---|---|---|---|---|---|
| $\theta$ = | 0 | -1 | 7 | 1 | 2 | 0 | 0 |
| $y_1$ = | -1 | -1 | 0 | 0 | 0 | 1/2 | 1/2 |
| $y_2$ = | 7 | 0 | -1 | 0 | 0 | -11/2 | -3/2 |
| $y_3$ = | 1 | 0 | 0 | -1 | 0 | -5/2 | -1/2 |
| $y_4$ = | 2 | 0 | 0 | 0 | -1 | 9 | 1 |
| $x_5$ = | 0 | -1/2 | 11/2 | 5/2 | -9 | 0 | 0 |
| $x_6$ = | 0 | -1/2 | 3/2 | 1/2 | -1 | 0 | 0 |

After 6 pivots, one returns to the same schema.

## 2.5. <u>On dropping degeneracy precautions</u>.

As is well known, it is customary in practice to ignore the degeneracy problem in many mathematical programming schemes. Quadratic programming is no exception. Here we wish to draw some attention to the fact that occasionally this can be done with absolute confidence. To this end, we consider the application of Keller's algorithm (minus degeneracy precautions) to a special type of problem.

Theorem 2. Keller's algorithm applied to the bisymmetric schema

$$\begin{array}{r|c|cc|} & 1 & x_I & y_J \\ \hline \theta = & 2\kappa & c^T & -b^T \\ \hline y_I = & c & D & A^T \\ x_J = & b & -A & E \\ \hline \end{array}$$

in which $b \geq 0$ and $E$ is positive definite, provided $J \neq \phi$, uses only in-pivots and out-pivots.

Proof. Suppose $J$ is vacuous. Let $y_s$ be the distinguished variable; then $x_s$ is the driving variable. Since $J = \phi$, $y_s$ is the only eligible blocking variable. If $d_{ss} = \partial y_s/\partial x_s \leq 0$, then $y_s$ does not block $x_s$. The procedure terminates with $\theta$ going to minus infinity. If $d_{ss} > 0$ then the in-pivot $\langle y_s, x_s \rangle$ brings about a bisymmetric schema with $J = \{s\}$ , $I = I - \{s\}$ , and $E = [1/d_{ss}]$.

Thus, we may assume $J \neq \phi$. Note that so long as E remains positive definite, there will be no exchange pivots, for these come about when some $x_t$ blocks an $x_s$ and $\partial x_t/\partial y_t = e_{tt} = 0$. Now clearly an out-pivot $\langle x_t, y_t \rangle$ preserves the positive definiteness of E unless $J = \{t\}$. Under the action of an in-pivot $\langle y_s, x_s \rangle$ , E is replaced by a principal transform of the positive definite matrix

$$\begin{bmatrix} d_{ss} & A^T_{\cdot s} \\ -A_{\cdot s} & E \end{bmatrix}$$

Hence E will be positive definite when it is nonvacuous. □

Corollary. Under the assumptions of Theorem 2, Keller's method is finite.

Proof. If circling occurs in Keller's method there must be at least two exchange-pivots. □

Remarks. Some related results should be mentioned in connection with Theorem 2. In his Ph.D. thesis [10], Keller notes that his method executes no exchange-pivots when D is positive semi-definite and E is positive definite, but the discussion there is not related to the degeneracy problem. Also, in Appendix I of [ 5 ], Cottle and Djang observe that the symmetric van de Panne-Whinston algorithm [13] applied to the least-distance problem studied by Wolfe [16] does not require a non-degeneracy assumption to assure its finiteness.

We have presented the application of Bland's double least-index rule to Keller's method (and therefore implicitly to the van de Panne-Whinston symmetric algorithm). There is reason to believe that such a rule could apply to other algorithms, but we do not pursue this possibility here.

BIBLIOGRAPHY


[1] D. AVIS and V. CHVATAL, Notes on Bland's pivoting rule, Centre de recherches mathematiques, Universite de Montreal, CRM-646, September, 1976.

[2] R.G. BLAND, New finite pivoting rules for the simplex method, CORE Discussion Paper 7612, June 1976. Also in Mathematics of Operations Research 2 (1977), 103-107.

[3] A. CHARNES, "Optimality and degeneracy in linear programming", Econometrica 20 (1952), 160-170.

[4] R.W. COTTLE, "Symmetric dual quadratic programs", Quarterly of Applied Mathematics 21, 237-243.

[5] R.W. COTTLE and A. DJANG, Algorithmic equivalence in quadratic programming I: A least-distance programming problem, Technical Report 76-26, Department of Operations Research, Stanford University.

[6] G.B. DANTZIG, Quadratic Programming: A variant of the Wolfe-Markowitz algorithms, Research Report 2 (1961), Operations Research Center, University of California at Berkeley.

[7] ___________, Linear Programming and Extensions. Princeton University Princeton, 1963.

[8] G.B. DANTZIG, A. ORDEN and P. WOLFE, "Generalized simplex method for minimizing a linear form under linear inequality restraints", Pacific J. Math. 5 (1955), 183-195.

[9] E.L. KELLER, "The general quadratic optimization problem", Mathematical Programming 5 (1973), 311-337.

[10] ___________, Quadratic Optimization and Linear Complementarity, Ph.D. Thesis, University of Michigan, 1969.

[11] A. MAHTHAY, "Optimality conditions for quadratic programming", Mathematical Programming 1 (1971), 359-365.

[12] K.T. MARSHALL and J.W. SUURBALLE, "A note on cycling in the simplex method", Naval Research Logistics Quarterly 16 (1968), 121-137.

[13] VAN DE PANNE and A. WHINSTON, "The symmetric formulation of the simplex method for quadratic programming", Econometrica 37, 507-527.

[14] T.D. PARSONS, "A combinatorial approach to convex quadratic programming", Ph.D. Thesis, Princeton University, Princeton, N.J. (1966). Revised in Linear Algebra and Its Applications 3 (1970).

[15] P. WOLFE, "A technique for resolving degeneracy in linear programming", J. Soc. Indust. Appl. Math. 11 (1963), 205-211.

[16] ________, "Algorithm for a least-distance programming problem", Mathematical Programming Study 1 (North Holland) 1974, 180-205.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>78-3 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>LEAST-INDEX RESOLUTION OF DEGENERACY IN QUADRATIC PROGRAMMING | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical Report |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Yow-Yieh CHANG and Richard W. COTTLE | | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-75-C-0267 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Department of Operations Research<br>Stanford University<br>Stanford, CA 94305 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>NR-047-143 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Operations Research Program -- ONR<br>Department of the Navy<br>800 N. Quincy Street, Arlington, VA 22217 | | 12. REPORT DATE<br>February 1978 |
| | | 13. NUMBER OF PAGES<br>19 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

DEGENERACY
QUADRATIC PROGRAMMING
LEAST INDEX RULES
KELLER'S ALGORITHM

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this study, ~~we~~ combines least-index pivot selection rules with Keller's algorithm for quadratic programming to obtain a finite method for processing degenerate problems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)